OAAS

Objects as a Service

$1T

Global business software and services market size by 2029[†]

No progress since 1980s

Software development technology is still stuck with languages and approaches to build software developed in '80s.

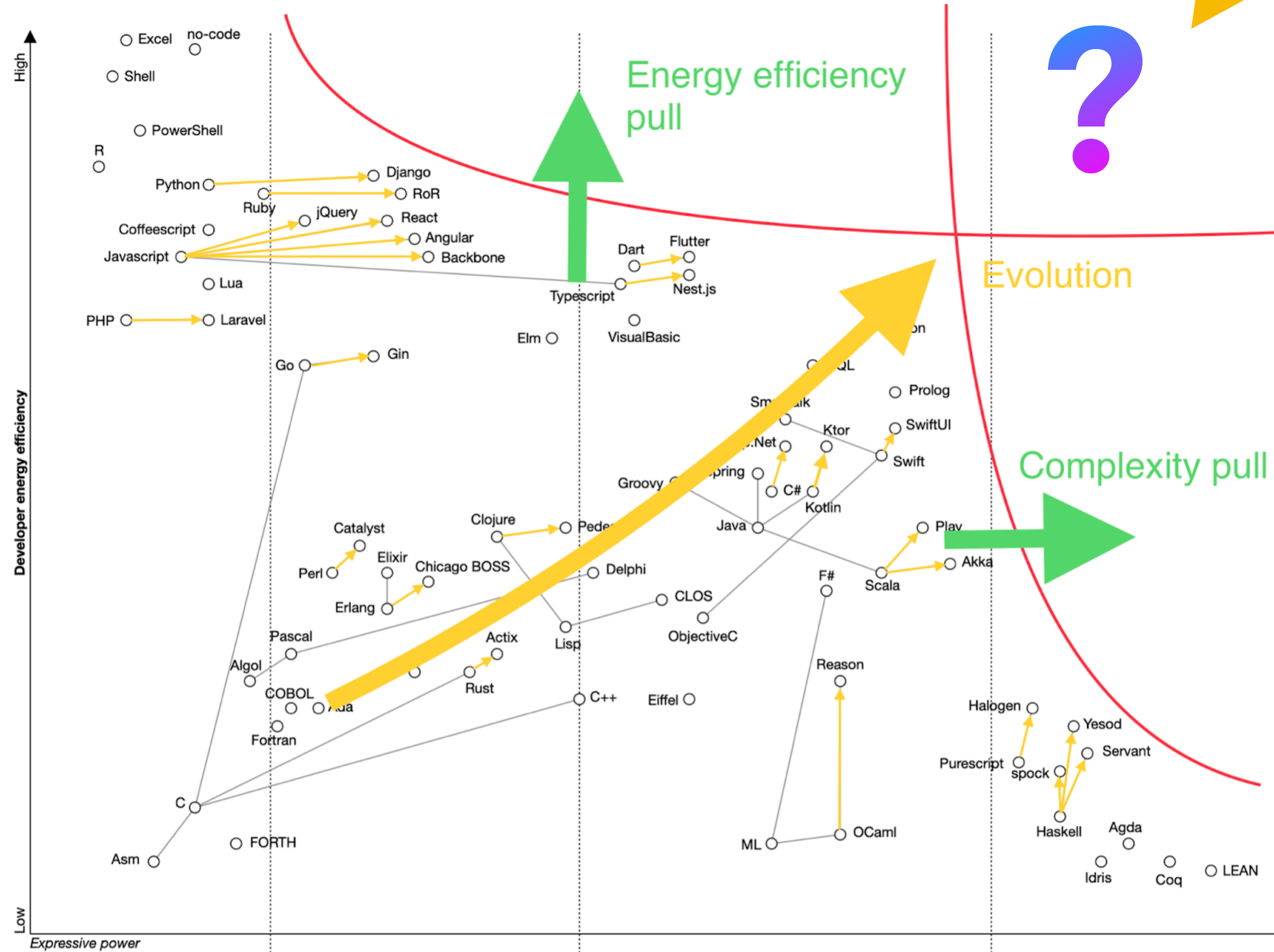More scalable horses, no automobiles.

Evolution →

Need for *expressive power* and human *energy efficiency* are the main drivers and fitness functions of evolution
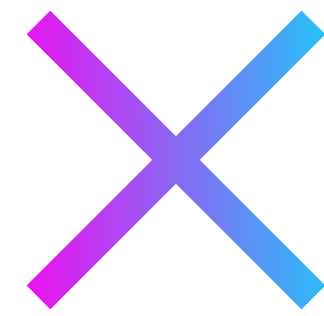
← Future

"Instead of innovating out from the present, what you want to do is invent the future from the future. Go out and live in the future and bring the future back."

— Alan Kay

AGI*

Human
language

Energy efficiency
pull

?

Evolution

Complex
systems

Complexity pull

* AGI will need better
software development
technology too

traversable.space/#essays/guts-d

High

Low

Developer energy efficiency

Expressive power

Excel   no-code
Shell
PowerShell
R
Python   Django
Ruby   RoR
Coffeescript   jQuery   React
Javascript   Angular
                        Backbone
Lua   Dart   Flutter
Typescript   Nest.js
PHP   Laravel
Elm   VisualBasic
Go   Gin
                QL
         Prolog
Smalltalk
         Ktor   SwiftUI
.Net   Swift
Groovy Spring   Kotlin
         C#
Java
Clojure   Peder   Play
Catalyst   Akka
Perl   Elixir   Chicago BOSS   Scala
Erlang   Delphi   F#
Pascal   CLOS
Algol   Actix   Lisp   ObjectiveC
COBOL   Ada
Rust   C++   Eiffel   Reason
Fortran
                Halogen   Yesod
                Purescript spock Servant
C
                Haskell   Agda
Asm   FORTH   ML   OCaml   Idris   Coq   LEAN
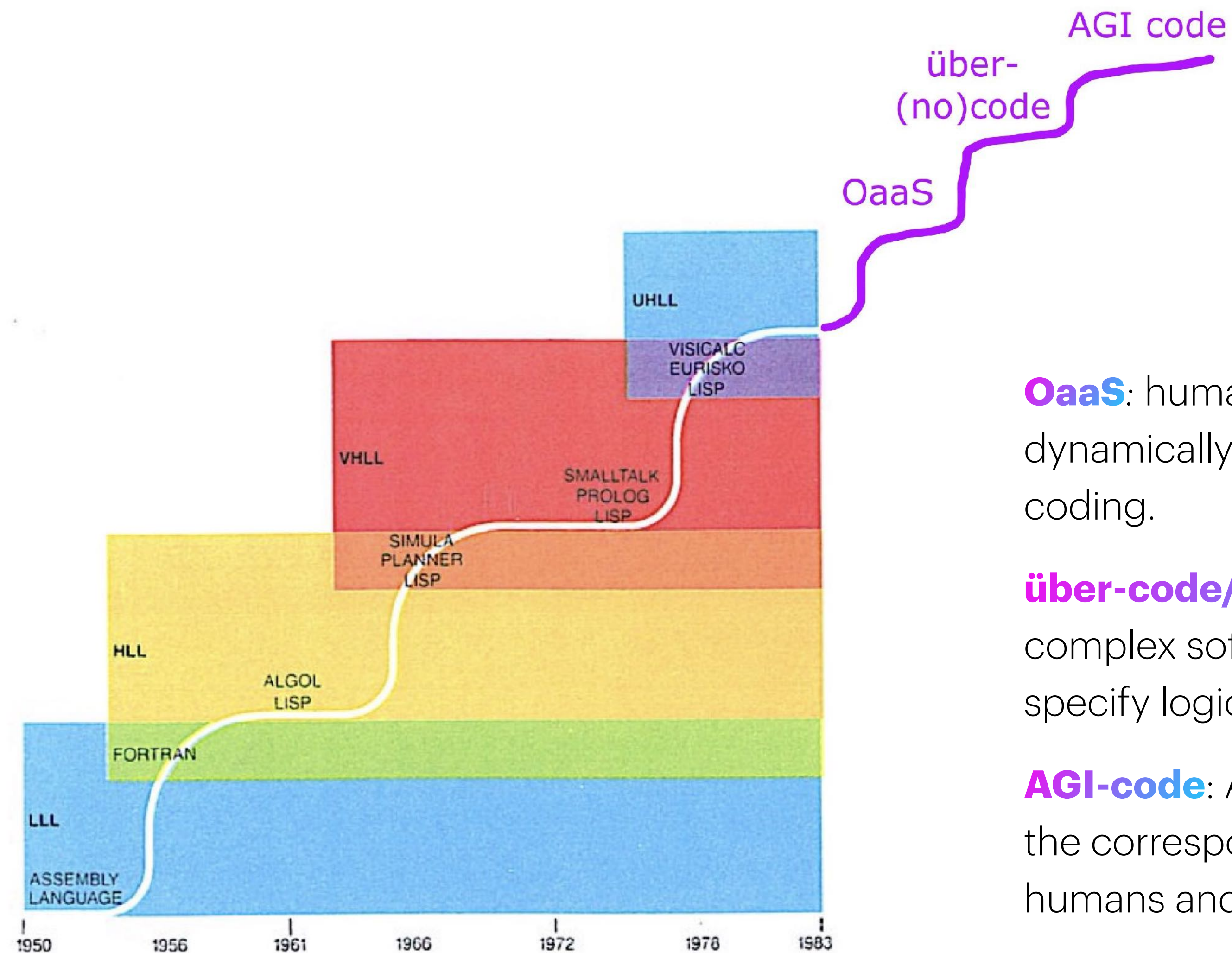
Complex Systems

✕

Human Language

Build complex, evolving software systems using human language to define

knowledge structure required to describe systems

OAAS →
über-no/code →
AGI-code

Build *systems* with no-code → build systems with *narratives* → build systems with *human language* → have *AGI* build systems

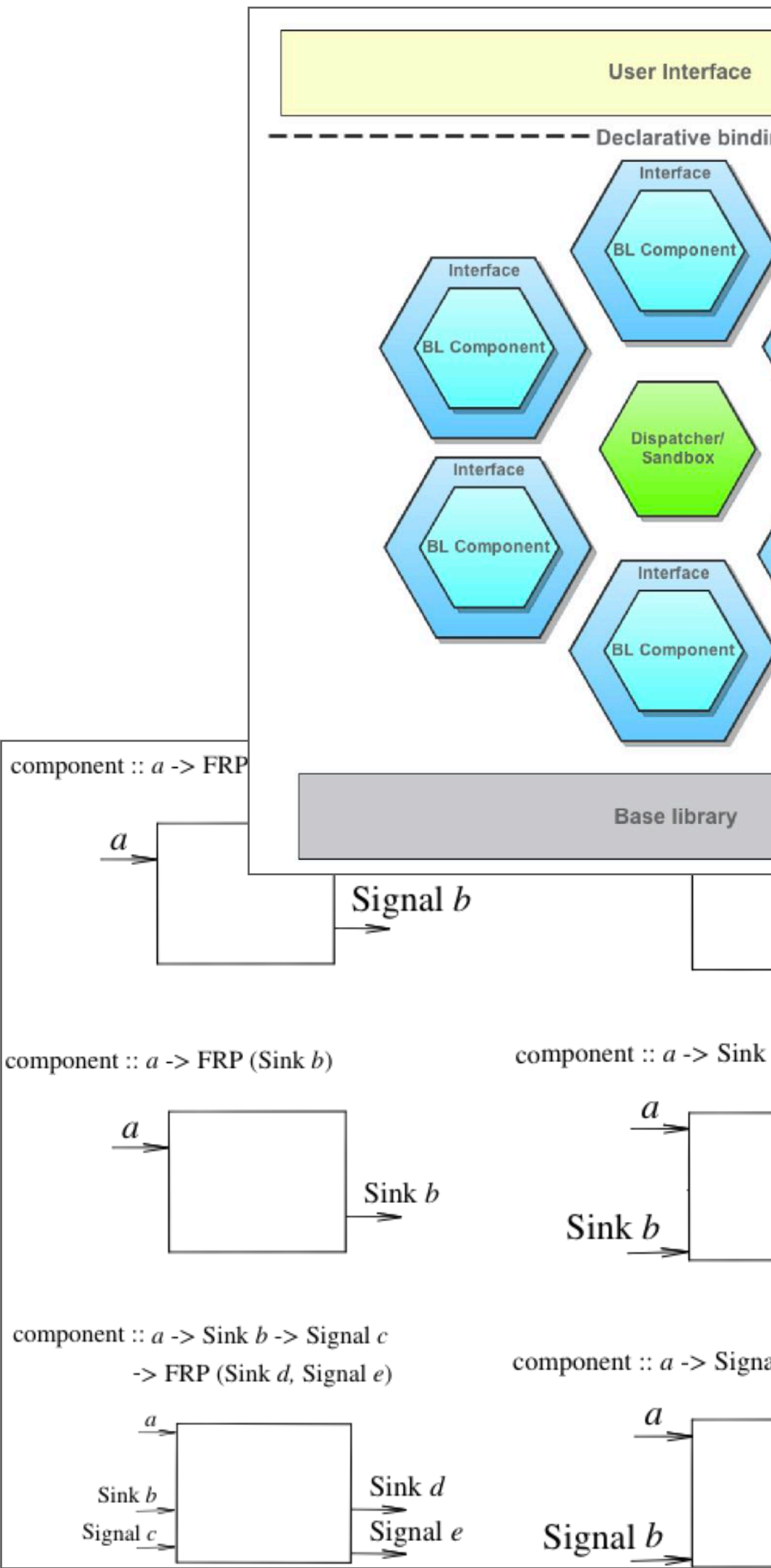**OaaS**: humans build complex software *systems* out of dynamically created and composed objects, without coding.

über-code/über-nocode: humans+*AI assistants* build complex software systems using human language to specify logical models, constraints and attractors.

AGI-code: AGI builds and updates logical models and the corresponding software by interacting with humans and the environment.

Proven with prototypes

Did enable building and evolving complex systems 10x faster,
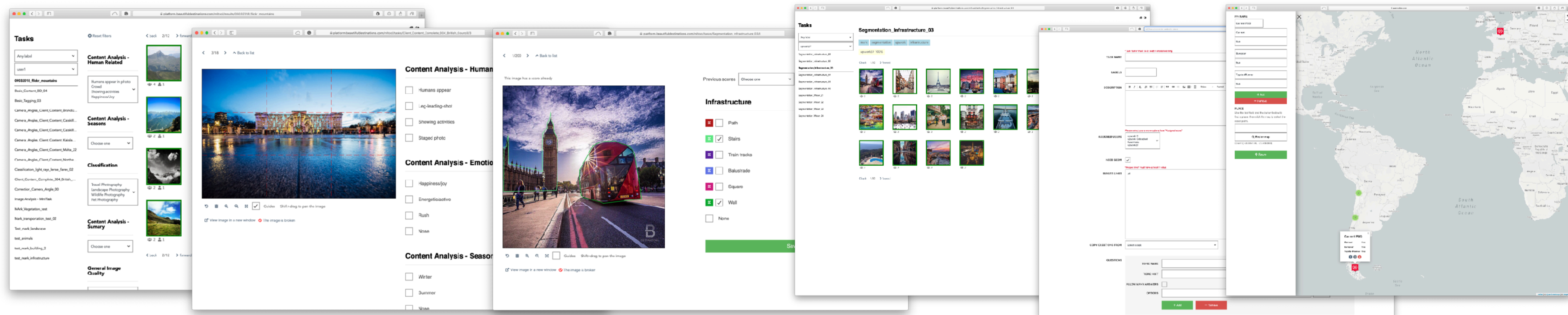
by non-programmers
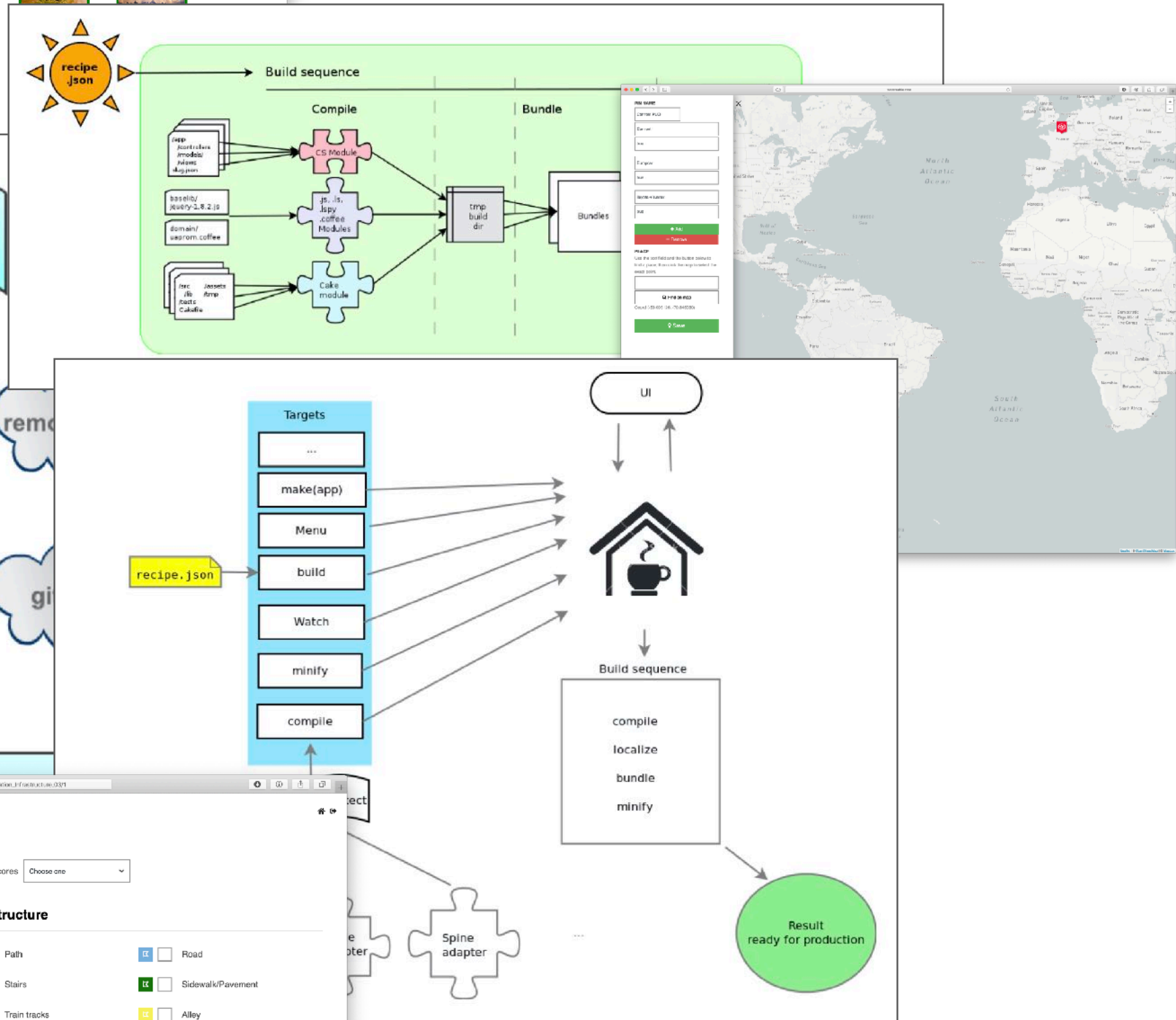
Built in 3 days

An application which is more powerful and feature-rich than the main product of a company with $20M funding after 2 years of development. Used in production in 3 days instead of a $18k contract with the aforementioned company.

New industry

Everyone able to step-by-step define a logical domain model is able to build software.

"Instead of innovating out from the present, what you want to do is invent the future from the future. Go out and live in the future and bring the future back."

"Computer science inverts the normal. In normal science, you're given a world, and your job is to find out the rules. In computer science, you give the computer the rules, and it creates the world."

"If you want to make money, don't make a startup, start an industry."

— Alan Kay

EUGENE NAUMENKO

eugene@traversable.space

https://traversable.space